BUI CGI – Gateway to the Internet

T he Common Gateway Interface (CGI) has been around for years, and anyone who has ever purchased a product off of the Internet has dealt with CGI first hand. In a nutshell, CGI refers to how a web server delegates the generation of dynamic HTML pages to a server-side program.

Our general purchasing experience on the Internet is a prime example of CGI, as the checkout process involves the customer adding items to a shopping cart, filling out a purchase form, and submitting the form. The form generally contains customer information like name and address, as well as purchasing information such as a credit card number and expiration date. When the customer fills in the form and clicks on the submit button, the web server executes the server-side CGI program referenced in the HTML page's ACTION attribute of the form. The CGI program then accesses the form's data and proceeds accordingly, eventually sending the customer a confirmation page. The gateway portion in this example is the interesting part - it is the connection between the data as it exists in the customer's web browser and the program on the web server that will eventually access and process the data.

Why use BUI Instead of Traditional CGI?
So what does all of this have to do with BBx® and BUI technology in particular? Because a BUI application is able to access CGI environment data, just like any other CGI

application, it is possible to leverage a BBj® BUI program as a reasonable alternative to the typical CGI Script/HTML page combo. This is exciting for all sorts of reasons, including:

- BBj is a familiar, straightforward language that produces rapid results
- BBj program development and debugging is much easier compared to most other CGI solutions
- BBj offers advanced controls such as masked input controls and popup calendars that simplify development

But one of the most tempting reasons to use a BUI program is that one can update existing portions or add new capabilities to an existing web system in an à la carte fashion – there is absolutely no need to completely rewrite the whole system. Because a BUI program has access to the CGI environment, it's possible to replace a single aspect of an existing system by simply changing the existing HTML page's form ACTION attribute to reference a BUI program instead of the legacy CGI script.

A Sample Use Case
Let's say your company has an existing web-based solution that utilizes a CGI language like Perl to gather form-based user information for customer feedback. The existing system is in need of an overhaul to add more features and flexibility to the survey capabilities. The original HTML page that gathers customer feedback still suffices, but the back-end CGI script > >

By Nick Decker
Engineering
Supervisor

needs modifications to further process the customer information and propagate it into an updated database structure. One solution would be to swap out the outdated Perl CGI program and slip a BBj BUI application in its place. Development and testing time would be decreased dramatically as the BUI program could access the existing business logic in your company's internal application. This means that a single code base is now responsible for processing customer information, instead of having separate systems written in different languages. The only trick is getting the customer information from the web-based form to the existing BBj processing routines.

The BUI CGI Demo Shows How It Is Done

The "BUI CGI" demo installed with BBj is a simplified version of our sample use case. When run, it first presents the user with an HTML page in their browser that uses a FORM element to gather customer feedback, as shown in **Figure 1**.
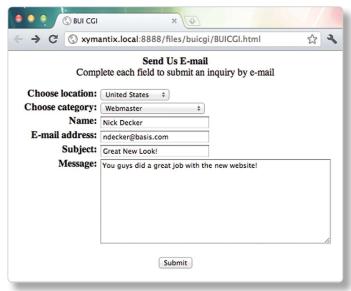

Figure 1. The BUI CGI's customer feedback HTML page

The HTML page is a straightforward web page served by the Jetty Web Server, included with the BBj product suite. Taking a look under the covers at the page's source code reveals the key FORM element, shown in **Figure 2**.

```
<form action="/apps/BUICGI" method="post">
```

Figure 2. The FORM element for the customer feedback HTML page

HTML FORM elements have an ACTION attribute that specify where to send the form's data when the user submits the form. Instead of referring to a Perl CGI script, this demo references a BBj BUI app instead. We've previously registered the BUI CGI program on our Web Server, and Enterprise Manager provides a quick way to copy the BUI app's URL into your clipboard, so it is a cinch to paste it into the HTML source code. The result of this change is that the Web Server launches our BUI program following the form submission.

The BUI program makes use of the CLIENTENV() function to retrieve the HTML FORM's data and store it into string variables for later use, as shown in **Figure 3**.

```
REM Get the information from the HTML form via clientenv
location$ = clientenv("location")
category$ = clientenv("category")
name$ = clientenv("name")
email$ = clientenv("email")
subject$ = clientenv("subject")
message$ = clientenv("message")
```

Figure 3. Using the CLIENTENV() function to retrieve FORM data

Now that the BUI program has access to all the FORM's data, it can CALL existing routines to process the customer feedback. The demo is a simplified version, though, so it creates a window with several controls to mimic the original HTML page and loads all the appropriate controls with the customer feedback. **Figure 4** shows the end result which is a "BUI-ized" version of the original HTML survey form.
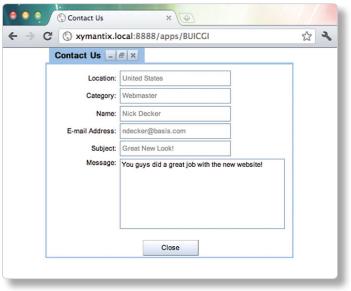

Figure 4. The BUI version of the feedback form loaded with the user's data

This example shows how easy it is to have an existing HTML page send all of its data to a BUI program. Our BUI demo resulted in a GUI program displayed on the end user's browser, but it could have just as easily redirected the browser to an existing submission confirmation page after processing the data.

Summary

Launching a BUI program is as easy as clicking on a link or pasting in a URL. Now you can use that URL in a brand new way – by specifying a BBj BUI program as the resultant ACTION for HTML form submissions. A BBj program is ready for the task, as it has complete access to all the HTML form's data, supports both GET and POST methods, and is a fast, robust programming language with excellent database integration. All of these features should make BUI your first choice for new Web development in the future. ■